UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/734,072	12/11/2000	Erik B. Christensen	146960.1/40062.72US01	3866

7590 05/17/2004

Homer L. Knearl
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
SHRADER, LAWRENCE J	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 05/17/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/734,072	CHRISTENSEN ET AL.	
	Examiner	Art Unit	
	Lawrence Shrader	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on March 17, 2004.

2. The Applicant's argument has been fully considered, but it was not persuasive.

3. Claims 1 – 30 remain rejected and repeated below.

Claim Objections

4. The objection to claims 21, 28, and 30 for including reference characters not enclosed within parentheses has been withdrawn in light of the amendments.

DETAILED ACTION

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1 – 6, 7 – 12, and 13 - 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fowlow et al., U.S. Patent 6,189,138 (hereinafter referred to as Fowlow) in view of Goldberg et al., U.S. Patent 6,571,232 (hereinafter referred to as Goldberg).

In regard to claim 1:

"dragging a graphical representation for a server processing resource from a server explorer module to a visual design surface module to add a processing item to a programmable data object being created in the visual design surface module;" Fowlow discloses a drag and drop function wherein a component representing software resources on a distributed object system (column 9, lines 45 – 47) is selected and dragged from one panel and dropped onto another panel being a worksheet (visual design surface) where the object component is manipulated in an application construction (column 11, lines 29 – 46).

"identifying data schema associated with the server processing resource added to the programmable data object;" Fowlow discloses a drag and drop function that allows a user to configure objects in a visual design surface giving the developer detailed information about the objects (column 12, lines 20 – 24), but does not specifically disclose identification of a data schema associated with a resource. However, Goldberg discloses identification of a database schema associated with a server resource (column 2, lines 9 – 37; e.g., Figure 3). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop feature for building applications in the Fowlow invention with the identification of a database schema in the Goldberg invention because the Fowlow invention discloses that detailed information about a resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically

include a data schema if a database resource was selected in a tiered distributed object system as taught by Goldberg (column 2, lines 17 – 37).

"creating a typed dataset containing the data structures corresponding to the data schema associated with the server processing resource;" Fowlow discloses a drag and drop function that allows a user to configure objects in a visual design surface giving the developer detailed information about the objects, but does not disclose a dataset containing data structures corresponding to the resource schema. However, Goldberg discloses generation of query objects based on a schema of an underlying database (column 2, lines 15 – 31; column 3, lines 13 – 19). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing a user to configure objects in a visual design surface giving the developer detailed information about the objects in the Fowlow invention with the generation of query objects based on a schema on an underlying database in the Goldberg invention because the Fowlow invention discloses that detailed information about a resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

"creating a command adapter to provide data transfer commands within the programmable data object between the programmable data object and the server processing resource;" Fowlow discloses that a distributed object system typically includes an object request broker to provide transport mechanisms for communications between a client and a server (column 14, lines 24 – 58; e.g. Figure 10).

"creating a data transfer connection between the programmable data object and the server processing resource." Fowlow discloses that a distributed object system typically includes an object request broker to provide transport mechanisms for communications between a client and a server (column 6, lines 16 – 21 and 30 – 37).

In regard to claim 2, incorporating the rejection of claim 1:

"...creating additional initialization methods to support the items added to the programmable data object." See Fowlow, column 11, lines 20 – 46.

In regard to claim 3, incorporating the rejection of claim 2:

"...editing the processing items within the visual design surface module; See Fowlow column 4, lines 18 – 26.

updating the processing items edited within visual design surface module; See Fowlow column 4, lines 35 – 43.

updating the typed dataset within visual design surface module; See Fowlow column 4, lines 26 – 34.

updating the command adapters within the visual design surface module; See Fowlow column 4, lines 26 – 34.

identifying any other processing items containing references to data structures and functions edited; See Fowlow column 4, lines 26 – 34.

updating the identified items containing references to data structures and functions edited to make all references consistent with each other." See Fowlow column 4, lines 26 – 34.

In regard to claim 4, incorporating the rejection of claim 3:

"..the items comprise properties and processing instruction source code that may be edited." See Fowlow column 4, lines 35 – 43.

In regard to claim 5, incorporating the rejection of claim 1:

"...inserting a database connection module that creates a data transfer connection between the programmable data object and a database when the dragged item is database table within the database." Fowlow teaches a means to insert connections among modules, but does not specifically teach that the dragged item is a database table. However, Goldberg teaches query objects of schemas including database tables (column 3, lines 37 – 54). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the insertion of connections among modules in a visual design surface as taught by Fowlow with the generation of query objects based on a schema on an underlying database, including tables in the Goldberg invention, because one skilled in the art would be motivated to include all relevant features of the database that require generation of code for an interface of a query object as taught by Goldberg (Abstract) in a visual tool designed to construct object-oriented applications as taught by Fowlow.

In regard to claim 6, incorporating the rejection of claim 5:

"a data connection object for creating and managing the data transfer connection between the programmable data object and the database; Fowlow teaches implementation of relationships among objects when the application is run (Abstract; column 4, lines 45 – 60)

a managed resource module for provides the data connection object with address and identification information to establish that data transfer connection; Fowlow discloses that a

distributed object system typically includes an object request broker to provide transport mechanisms for communications between a client and a server (column 6, lines 16 – 21 and 30 – 37).

a persistent data storage for maintaining this address and identification information used by the managed resource module." See Fowlow Figure 1.

In regard to claim 7:

Claim 7 (a system claim) is rejected for the same reasons put forth in the rejection of claim 1 (a corresponding method claim) with the following differences also found in the Fowlow reference:

"a memory module; (Figure 3)

a user interface module; (Figure 4)

a mass storage system; (Figure 1)

In regard to claim 8, incorporating the rejection of claim 7:

Claim 8 (a system claim) is rejected for the same reasons put forth in the rejection of claim 2 (a corresponding method claim).

In regard to claim 9, incorporating the rejection of claim 8:

Claim 9 (a system claim) is rejected for the same reasons put forth in the rejection of claim 3 (a corresponding method claim).

In regard to claim 10, incorporating the rejection of claim 9:

Claim 10 (a system claim) is rejected for the same reasons put forth in the rejection of claim 4 (a corresponding method claim).

In regard to claim 11, incorporating the rejection of claim 10:

Claim 11 (a system claim) is rejected for the same reasons put forth in the rejection of claim 5 (a corresponding method claim).

In regard to claim 12, incorporating the rejection of claim 11:

Claim 12 (a system claim) is rejected for the same reasons put forth in the rejection of claim 6 (a corresponding method claim).

In regard to claim 13:

Claim 13 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 1 (a corresponding method claim).

In regard to claim 14, incorporating the rejection of claim 13:

Claim 14 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 2 (a corresponding method claim).

In regard to claim 15, incorporating the rejection of claim 14:

Claim 15 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 3 (a corresponding method claim).

In regard to claim 16, incorporating the rejection of claim 15:

Claim 16 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 4 (a corresponding method claim).

In regard to claim 17, incorporating the rejection of claim 13:

Claim 17 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 5 (a corresponding method claim).

In regard to claim 18, incorporating the rejection of claim 17:

Claim 18 (a computer program product claim) is rejected for the same reasons put forth in the rejection of claim 6 (a corresponding method claim).

In regard to claim 19:

"a server explorer module for presenting one or more processing resources present on a server to a programmer for use in creating a programming object class; See the Fowlow Abstract, Figure 5, and column 11, lines 22 – 35).

"a visual design surface module for performing the operations associated with creating, editing, and saving the programming object, the visual design surface module comprises:

a drag/drop module for enabling a programmer to select a server resource from the server explorer module and place the selected server resource within a data object on the visual design surface module; Fowlow discloses a drag and drop function wherein a component representing software resources on a distributed object system (column 9, lines 45 – 47) is selected and dragged from one panel and dropped onto another panel being a worksheet (visual design surface) where the object component is manipulated in an application construction (column 11, lines 29 – 46).

a command adapter function generation module for generating a data processing object associated with the drag and drop of a server processing resource; The Fowlow invention generates code corresponding to the connection (resource) in the design surface (column 11, lines 37 – 63).

a typed dataset generation module for generating typed dataset object associated with the drag and drop of a server processing resource; Fowlow discloses a drag and drop function that allows a user to configure objects in a visual design surface giving the developer detailed information about the objects, but does not disclose a dataset containing data structures corresponding to the resource schema. However, Goldberg discloses generation of query objects based on a schema on an underlying database (column 2, lines 15 – 31; column 3, lines 13 – 19). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing a user to configure objects in a visual design surface giving the developer detailed information about the objects in the Fowlow invention with the generation of query objects based on a schema on an underlying database in the Goldberg invention because the Fowlow invention discloses that detailed information about an object resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

an init generation module for generating the set of data processing functions and methods associated with the drag and drop of a server processing resource; The Fowlow invention generates code corresponding to the connection (resource) in the design surface (column 11, lines 37 – 63).

a properties edit module for retrieving the properties and source code for the various objects within the visual design surface module for editing." See Fowlow column 4, lines 18 – 26 teaching a composition workspace for editing.

In regard to claim 20, incorporating the rejection of claim 19:

"...the drag and drop module comprises:

an explorer interface module to select a server resource from the server explorer module and place it within a data object within the visual design surface module; Fowlow discloses the selection of a component from one module and placing it on a visual design surface into a part representing an object (Abstract, Figure 5).

a user interface module to perform the visual display and command input operations associated with the drag/drop operation; In the Fowlow invention, the user interfaces with the visual display including inputs associated with the drag and drop operation (Figure 5, column 11, lines 27 – 46).

a class generation module to cause the visual design surface module to perform the operations to complete the drag/drop process of a server resource onto the visual design surface module." See Fowlow column 11, lines 1 – 28.

In regard to claim 21, incorporating the rejection of claim 19:

"..the drag/drop module further causes the other processing modules in the visual design surface module 402 to perform their operations to complete the drag/drop process of a server resource onto the visual design surface module." See Fowlow column 11, lines 29 – 56.

In regard to claim 22, incorporating the rejection of claim 19:

"...the command adapter function generation module comprises:

a GetDS module for generating a GetDataSet function that fills a typed dataset with data obtained from a corresponding database;

an updateDS module for generating an UpdateDataSet function that updates a database using the data stored within the typed dataset." Fowlow does dot teach database access functions, but Goldberg teaches a query object with database query commands (column 2, lines 38 – 43), including select (getDS) and update (updateDSs). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing a user to configure objects in a visual design surface giving the developer detailed information about the objects in the Fowlow invention with the generation of query objects based on a schema of an underlying database in the Goldberg invention, because the Fowlow invention discloses that detailed information about an object resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema and database queries if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

In regard to claim 23, incorporating the rejection of claim 19:

"...the command adapter function generation module further accepts an updated command adapter module that has been edited by the properties edit module and generates the updated source code for the functions within the command adapter modules." See Fowlow column 11, lines 29 – 56.

In regard to claim 24, incorporating the rejection of claim 19:

"...the typed dataset generation module comprises:

a Table Schema module for generating the table records from the database schema within the dataset object; and

a Relations module for generating the relationship data for the fields within the records within the dataset based upon the corresponding relationship data from the database; Fowlow discloses a drag and drop function that allows a user to configure objects in a visual design surface giving the developer detailed information about the objects, but does not disclose a dataset containing data structures corresponding to the resource schema or relations for the fields within the dataset. However, Goldberg discloses generation of query objects based on a schema on an underlying database (Abstract; column 2, lines 17 – 37) generation of relationship data for the fields (column 3, lines 40 – 45). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing a user to configure objects in a visual design surface giving the developer detailed information about the objects in the Fowlow invention with the generation of query objects based on a schema of an underlying database in the Goldberg invention because the Fowlow invention discloses that detailed information about an object component available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

a Views module for generating the database views data for the records within the dataset based upon the corresponding views data from the database." Fowlow does not disclose a view of database records, but Goldberg discloses a view of database records in constructing an SQL query (column 37, lines 51 – 65; Figure 8). Therefore, it would have been obvious to one skilled

in the art at the time the invention was made to combine the drag and drop function allowing a user to configure objects in a visual design surface giving the developer detailed information about the objects in the Fowlow invention, with the generation of query objects based on a schema on an underlying database in the Goldberg invention in order to create views of the database, because the Fowlow invention discloses that detailed information about a resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema to construct views for queries if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

In regard to claim 25, incorporating the rejection of claim 19:

"... the typed dataset generation module further accepts an updated typed dataset module that has been edited by the properties edit module and generates the updated source code for the functions within the typed dataset module." Fowlow generates code for the application software under construction (Abstract), but does not accept an updated typed dataset module. However, Goldberg discloses query objects that encapsulate a specific query language and the database organization or schema, including typing. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing code generation for the application software under construction in the Fowlow invention, with the generation of query objects based on a schema on an underlying database in the Goldberg invention in order to create views of the database with a query language, because the Fowlow invention discloses that detailed information about a resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art

would logically include a data schema to construct queries if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

In regard to claim 26, incorporating the rejection of claim 19:

"...the init generation module comprises:

an Init Function module for generating the processing functions and methods within the programmable data object associated with the command adapter modules;

an InitDataSet module for generating the processing functions and methods within a programmable data object module associated with the type dataset class;

an InitConnection module for generating the processing functions and methods within the programmable data object associated with the data transfer connection between the programmable data object and the database."

The Fowlow invention generates code corresponding to the connection (resource) in the design surface (column 11, lines 37 – 63) but does not teach using a data set. However, the Goldberg invention generates queries in a specific query language (column 2, lines 17 – 64). In order to develop a query it would be inherently necessary to initialize a function (a query object) and the dataset and establish a connection to the database. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop function allowing code generation for the application software under construction in the Fowlow invention, with the generation of query objects based on a schema on an underlying database in the Goldberg invention in order to create views of the database with a query language, because the Fowlow invention discloses that detailed information about a selected object component available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface),

and one skilled in the art would logically include a data schema to construct queries if a database resource was selected in a tiered distributed object system as taught by Goldberg (Abstract; column 2, lines 17 – 37).

In regard to claim 27, incorporating the rejection of claim 19:

"...the init generation module further accepts an updated Code Generated Method module that has been edited by the properties edit module and generates the updated source code for the functions within the command adapter modules." Fowlow discloses an editing module available to update code for the interface functions (column 13, lines 17 – 30).

In regard to claim 28, incorporating the rejection of claim 19:

"...the properties edit module comprises:

a Properties Retrieval module for retrieving the properties and source code for the various objects within the visual design surface module 402 for editing; Fowlow teaches that properties and code are retrieved within the visual design surface for editing (column 13, lines 17 – 30).

a User Interface module for presenting the properties and source code to a programmer, and accepting edits from the programmer; See Fowlow column 13, lines 17 – 30.

a class update module for updating the edited object." See Fowlow (column 13, lines 12 – 30).

In regard to claim 29, incorporating the rejection of claim 19:

"...the class update module further:

identifies all other items within the visual design surface module that are affected by the changes made to the edited item;

updates the identified items consistent with the edits made to the edited item."

See Fowlow wherein all items affected by an update are identified and updated (column 11, lines 1 – 63).

In regard to claim 30, incorporating the rejection of claim 19:

"...the properties edit module further:

identifies all other objects within the visual design surface module 402 that are affected by the changes made by the update operations;

updates the identified objects."

See Fowlow wherein all objects affected by an update are identified and updated (column 11, lines 1 – 63).

Response to Arguments

7. Applicant's arguments filed on 3/17/2004 have been fully considered but they are not persuasive.

The Applicant has argued:

> "The rejection states that Fowlow fails to suggest (i) identifying data schema associated with a server processing resource, and (ii) creating a typed dataset containing the data structures corresponding to the data schema associated with the server processing resource, as recited by claim 1. The rejection cites Goldberg as suggesting such steps and states that it would have been obvious to one skilled in the art to combine the disclosures of Fowlow and Goldberg. This combination is respectfully traversed because there is no suggestion or motivation to combine Fowlow and Goldberg, and it is

therefore respectfully suggested that the rejection fails to provide a prima facie case of obviousness.

The possible sources. for motivation to combine references are: (a) the nature of problem being solved; (b) the teaching of the prior art; and (c) the knowledge of one skilled in the art. MPEP 2143.01. It is respectfully suggested that, in the present application, none of these sources provide motivation to combine Fowlow and Goldberg for at least the following reasons.

Specifically, Fowlow is directed at the problem of combining objects distributed across a network and does not even disclose or reference a database, while Goldberg is directed specifically at database access. Therefore, the problems being solved by Fowlow and Goldberg are different. Further, neither the disclosure of Fowlow nor Goldberg suggests such a combination for at least the same reasons. In addition, there is no suggestion that one of ordinary skill in the art would combine a Fowlow, a reference dealing with the combination of distributed objects, with Goldberg, a reference directed at providing database access."

Examiner's response:

The Office agrees with the Applicant with regard to the content of MPEP 2143.01, but disagrees that the MPEP citation has not been properly applied in the previous Office Action. Fowlow accesses network objects as shown in Figure 4 (see also column 9, line 32 to column 10, line 29). It was generally known by one skilled in the art that until roughly the mid-1990s, when software programs were created using object-oriented design, the designs operated mainly in the memory of single machines. Design methods were necessary for adapting the software for network use and saving the software in secondary storage. Distributed object technology for networks was developed to solve this problem, and object-oriented database (ODBMS) technology, which saves the objects directly in secondary storage, was developed. One skilled in the art would have understood that the access of network objects in the Fowlow invention would include database objects. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the drag and drop feature for building applications in the Fowlow invention, which is installed on a distributed object system, with the identification of

a database schema in the distributed object database system disclosed in Goldberg because the Fowlow invention discloses that detailed information about a distributed object system resource is available to the developer (column 9, lines 54 – 55) in the worksheet (a visual design surface), and one skilled in the art would logically include a data schema if a database resource was selected in a tiered distributed object system as taught by Goldberg (column 2, lines 17 – 37).

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

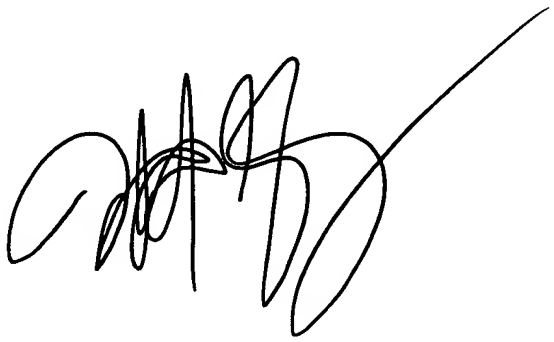
9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (703) 305-8046. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2124

14 May 2004



TODD INGBERG
PRIMARY EXAMINER